Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

CRT-2006-14

MODELS AND BRANCH-AND-CUT ALGORITHMS FOR THE STEINER TREE PROBLEM WITH REVENUES, BUDGET AND HOP CONSTRAINTS

by

Alysson M. Costa
Jean-François Cordeau
Gilbert Laporte

July 2006

Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal QC H3C 3J7 Canada
Téléphone: (514) 343-7575 / Télécopieur: (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca

HEC ÉCOLE POLYTECHNIQUE MONTRÉAL Université de Montréal

PUBLICATION
CRT-2006-14

MODELS AND BRANCH-AND-CUT ALGORITHMS FOR THE STEINER TREE PROBLEM WITH REVENUES, BUDGET AND HOP CONSTRAINTS

by

Alysson M. Costa[1]
Jean-François Cordeau[1]
Gilbert Laporte[1]

This research was supported in part by the Brazilian Federal Agency for Post-Graduate Education – CAPES, and by the Natural Sciences and Engineering Research Council of Canada (NSERC).

1. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7, and Canada Research Chair in Distribution Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7 (alysson, cordeau, gilbert@crt.umontreal.ca)

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2006

106-06867/

Abstract

The Steiner tree problem with revenues, budget and hop constraints is a variant of the Steiner tree problem where, besides the costs associated with arcs, there are also revenues associated with the vertices. Budget and hop constraints impose limits on the total cost of the network and on the number of edges between any vertex and the root, respectively. This article introduces and compares several mathematical models for this problem and describes two branch-and-cut algorithms which solve to optimality instances with up to 500 vertices and 625 edges.

Keywords: network design, Steiner tree problem, budget, branch-and-cut, hop constraints.

Résumé

Le problème de Steiner avec revenus, budget et contraintes d'éloignement est une variante du problème de Steiner où, en plus des coûts associés aux arcs, des revenus sont associés aux sommets. Des contraintes de budget et d'éloignement imposent des limites sur le coût total du réseau et sur le nombre d'arêtes entre un sommet quelconque et la racine du réseau. Cet article compare plusieurs modèles mathématiques pour ce problème et décrit deux procédures de séparation et évaluation qui résolvent à l'optimalité des instances comportant jusqu'à 500 sommets et 625 arêtes.

Mots-clés: conception de réseaux, problème de Steiner, budget, séparation et évaluation, contraintes d'éloignement.

1 Introduction

Several real-life decision situations can be described as the problem of determining a least cost network spanning all or some of the vertices of a graph. The most widely known cases include the *Steiner Tree Problem* (STP) and the *Minimum Spanning Tree Problem* (MSTP). These two problems are described as follows. Let $G = (V, E)$ be a graph with vertex set $V = \{1, \ldots, n\}$, where vertex 1 is the *root* vertex, and edge set $E = \{e = (i, j) : i, j \in V, i < j\}$, where each edge $e \in E$ has an associated cost c_e. The set V is partitioned into a set of *mandatory* vertices (including the root) and a set of *Steiner* vertices. The STP consists of determining a minimum cost subgraph spanning all mandatory vertices and possibly some Steiner vertices (see, for instance, Courant and Robbins, 1941; Gilbert and Pollak, 1968; and Hwang et al., 1992). The MSTP is a special case of the STP for which all vertices are mandatory. Unlike the MSTP, the STP is NP-hard (Garey et al., 1977).

Steiner Tree Problems with Revenues (STPR) are an important generalization of the classical STP. In the STPR, in addition to the costs associated with the edges, there is also a revenue $r_i \geq 0$ associated with each vertex i. The goal is to determine a cost minimizing or revenue maximizing subtree subject to constraints. This article deals with a particular case of the STPR with two main features. We are first interested in the *Steiner Tree Problem with Revenues and Budget* (STPRB), where the goal is to maximize the collected revenue while respecting an upper limit on the total network cost. We also consider hop constraints which limit the number of edges between any vertex in the solution and the root vertex to an upper limit equal to h. We call the resulting problem the *Steiner Tree Problem with Revenues, Budget and Hop Constraints* (STPRBH).

Several authors (see, e.g., Ljubić et al., 2005; Lucena and Resende, 2005) have studied a related version of the STPR where the goal is to maximize the difference between the collected revenues and the edge costs. This problem is usually called the *Steiner Tree Problem with Profits* (STPP). Since, in our case, the goal is to maximize the collected revenue while respecting a limit on the total cost (and not to maximize the difference between the collected revenue and the network cost), we use the term "revenue" in order to avoid confusion.

Our motivation for studying the STPRBH is twofold. Concerning the budget constraint, it has been observed that this problem has received very little attention in the literature despite the fact that these constraints are often present in real-life situations (Costa et al., 2006). The hop constraints in turn, are often used in telecommunications applications. They guarantee that the probability of service failure in a vertex will not exceed a given threshold $1 - (1 - \pi)^h$, where π is the probability of failure of any link and h is the number of allowed hops. These constraints also restrict the maximum transmission delay in telecommunication networks (Gouveia, 1996). Finally, Voß (1999) mentions a different motivation for considering Steiner trees with hop constraints: when modeling certain lot sizing problems as Steiner tree problems, hop constraints limit the number of periods some goods can be held in stock.

Steiner Tree Problems with Profits have received considerable attention by operations researchers (Costa et al., 2006). The first studies go back to Segev (1987) who implemented simple greedy heuristics and a Lagrangian lower bounding procedure. Bienstock et al. (1993) have proposed a heuristic with worst case performance ratio of 3, while Goemans and Williamson (1995), Johnson et al. (2000) and Cole et al. (2001) have developed approximation algorithms. More recent research has focused on lower bounding procedures and metaheuristics. Canuto et al. (2001) have proposed a multi-start algorithm with local search and a variable neighborhood search post-optimization phase. Cunha et al. (2003) have developed a Lagrangian relaxation algorithm with dynamically generated cutting planes, while Klau et al. (2004) have proposed a hybrid exact-memetic algorithm where the final population of the evolutionary approach is used to construct a reduced instance that will go through an exact optimization phase. Two recent articles have applied branch-and-cut algorithms to the STPP. Lucena and Resende (2004) have obtained interesting results based on a separation of generalized subtour constraints. These results have later been improved by Ljubić et al. (2005) who have dealt with an alternative formulation and proposed a separation procedure to find violated connectivity constraints.

Despite the considerable amount of research dedicated to the STPP, very little work has been done on the STPRB. Indeed, we are only aware of the article of Johnson et al. (2000) who proposed an approximation algorithm with limited practical interest. This lack of research is particulary intriguing due to the fact that budgetary constraints are often present in real-life situations. Concerning the hop constraints no author has, to our knowledge, ever considered their inclusion (or the inclusion of any other notion of reliability) in conjunction with the STPP. These constraints have only been considered in more classical contexts as the MST and STP. Gouveia (1996) and Gouveia and Requejo (2001) have proposed Lagrangean relaxation lower bounding approaches for the MST with hop constraints and were able to solve instances on complete graphs of up to 60 vertices. Dahl et al. (2006) present a general framework for modeling hop constrained MST problems. Finally, Voß (1999) has presented a mathematical formulation and a tabu search for the STP with hop constraints.

We study several mathematical models for the STPRBH and we propose two branch-and-cut algorithms. Our choice of branch-and-cut methods is motivated by the good results obtained by Lucena and Resende (2004) and Ljubić et al. (2005) in the context of the pure STPP. The first algorithm relies on the initial relaxation of connectivity and hop constraints, while the second relaxes a set of linking constraints on disaggregated variables. In both cases, the constraints are dynamically included in the model when found to be violated. Computational experiments show that the hop constraints make the problem highly difficult and that the choice of the most efficient strategy depends on the maximum number of hops.

The remainder of this paper is organized as follows. Section 2 proposes mathematical models for the STPRBH. In Section 3, we present the two branch-and-cut algorithms. Section 4 reports the computational results and the paper ends with some conclusions in Section 5.

2 Formulations

We introduce four formulations for the STPRB. In Section 2.1, we adapt subtour elimination constraints and connectivity constraints to construct an undirected and a directed formulation for the STPRBH, respectively. The resulting formulations contain a limited number of variables but an exponential number of constraints. A different approach is used in Section 2.2, where we modify a formulation for the STP with hop constraints presented by Voß (1999) to the case with revenues. The formulation obtained is relatively compact but also weak and is therefore strengthened with the help of additional variables. Contrary to formulations based on subtour elimination constraints or connectivity constraints, the formulation with additional variables contains a large number of columns but a polynomial number of rows.

Gouveia (1996) also presents a formulation belonging to a different family and based on multicommodity flow variables for MST problems with hop constraints, which could also be adapted to the STPRBH. A variation of the multicommodity flow formulation has been proposed by Gouveia (1998) and Gouveia and Requejo (2001), inspired by the idea of variable redefinition. This later formulation is a combination of an exact solution for the hop-constrained shortest path problem and the original multicommodity flow formulation. In both cases, the authors have proposed the use of Lagrangean relaxation methods for the solving of the formulations since computational tests have shown that even their linear relaxations pose a real challenge. For this reason, formulations of this type are not considered in this work. This study is therefore based on the first four formulations and on their combinations.

2.1 Dantzig-Fulkerson-Johnson formulations

Several mathematical formulations have been developed for the STPP and can be adapted to the STPRB. The following model is based on the classical subtour elimination constraints introduced by Dantzig et al. (1954) for the *Travelling Salesman Problem* (TSP). Let x_{ij} and y_i be binary variables associated with edges $(i,j) \in E$ and vertices $i \in V$, respectively. Variable y_i is equal to 1 if vertex i belongs to the solution ($y_1 = 1$), and to 0 otherwise. Likewise, variable x_{ij} is equal to 1 if edge (i,j) belongs to the solution, and to 0 otherwise. For $S \subseteq V$, define $E(S)$ as the set of edges with both end vertices in S. Let also $P = (i_1 = 1, \ldots, i_\ell)$ denote a path originating at the root node and containing ℓ vertices. Finally, define $\mathcal{P}_h$ as the set of paths P in G with length $\ell \geq h+2$. Each path $P \in \mathcal{P}_h$ contains at least $h+1$ edges and thus violates the hop constraint. The STPRBH can then be written as:

Undirected Dantzig-Fulkerson-Johnson (UDFJ) formulation

$$\text{Maximize} \sum_{i \in V} r_i y_i \tag{1}$$

subject to

$$\sum_{(i,j)\in E} x_{ij} = \sum_{i\in V} y_i - 1, \tag{2}$$

$$\sum_{(i,j)\in E(S)} x_{ij} \leq \sum_{i\in S\setminus\{k\}} y_i, \qquad k \in S \subseteq V,\ |S| \geq 2, \tag{3}$$

$$\sum_{(i,j)\in E} c_{ij} x_{ij} \leq b, \tag{4}$$

$$\sum_{t=2}^{\ell} x_{i_{t-1},i_t} \leq h, \qquad P = (i_1 = 1, \ldots, i_\ell) \in \mathcal{P}_h, \tag{5}$$

$$x_{ij} \in \{0,1\}, \qquad (i,j) \in E, \tag{6}$$

$$y_1 = 1, \tag{7}$$

$$y_i \in \{0,1\}, \qquad i \in V\setminus\{1\}. \tag{8}$$

The objective function maximizes the revenue of the spanned vertices. Constraint (2) forces the presence of $v-1$ edges, connecting the v spanned vertices. Constraints (3) are generalized subtour elimination constraints. These are stronger than the classical subtour elimination constraints used in the TSP formulation in which the right-hand side is $|S|-1$. Note that for subsets S formed only by spanned vertices, constraints (3) reduce to the classical subtour elimination constraints. Constraint (7) forces the presence of the root vertex in the solution. Constraints (4) and (5) are specific to the STPRBH. Constraint (4) is the budget constraint which forces the total network cost not to exceed the budget b, while constraints (5) guarantee that there are no more than h hops between the root vertex and any other vertex in the solution.

A version of model (1)–(8) with no budget and no hop constraints was used by Lucena and Resende (2004) and follows from an extended formulation for the STP proposed by Lucena (1991), Goemans (1994) and Margot et al. (1994). It is interesting to observe that in the absence of hop constraints, the x variables associated with edges need not be declared as integers. Indeed, when the y variables are equal to 0 or 1, constraints (2) and (3) define the convex hull of the characteristic vectors of the spanning trees on the subgraph of G induced by the selected vertices (Margot et al., 1994).

Formulation (1)–(8) is a straightforward model for the STPRBH. An equivalent model can be developed based on constraints which ensure the existence of a path between the root vertex and all other selected vertices. This has been done in the context of the STPP by Ljubić et al. (2005). Consider an arc set A containing two directed arcs (i,j) and (j,i) for each edge $(i,j) \in E$. A directed rooted tree is called a *Steiner arborescence*. Given a directed set of arcs, the STPRBH can be written as the problem of finding a Steiner arborescence spanning a subset Y of vertices with maximum revenue $\sum_{i\in Y} r_i$.

Directed Dantzig-Fulkerson-Johnson (DDFJ) formulation

$$\text{Maximize} \sum_{i \in V} r_i y_i \tag{9}$$

subject to

$$\sum_{(j,i) \in A} x_{ji} = y_i, \qquad i \in V \backslash \{1\}, \tag{10}$$

$$\sum_{(i,j) | i \in S, j \in V \backslash S} x_{ij} \geq y_k, \qquad S \subset V, 1 \in S, k \in V \backslash S, \tag{11}$$

$$\sum_{(i,j) \in A} c_{ij} x_{ij} \leq b, \tag{12}$$

$$\sum_{t=2}^{\ell} x_{i_{t-1}, i_t} \leq h, \qquad P = (i_1 = 1, \ldots, i_\ell) \in \mathcal{P}_h, \tag{13}$$

$$x_{ij} \in \{0, 1\}, \qquad (i,j) \in A, \tag{14}$$

$$y_1 = 1, \tag{15}$$

$$y_i \in \{0, 1\}, \qquad i \in V \backslash \{1\}. \tag{16}$$

The objective function (9) maximizes the collected revenue. Constraints (10) guarantee that each selected vertex has an indegree of one. Constraints (12) and (13) guarantee that the budgetary limit and the hop constraints are respected. Constraint (15) forces the presence of the root vertex in the solution. Finally, constraints (11) guarantee that the selected vertices are connected and are therefore called *connectivity constraints*. Fischetti (1991) has shown that these constraints can be rewritten as a directed version of the subtour elimination constraints (3), while Chopra and Rao (1994) have proved that the directed subtour elimination constraints dominate their undirected equivalent and, for this reason, model (9)–(16) is preferable to model (1)–(8).

An interesting valid inequality for the STPRB can be borrowed from the STP:

$$x_{ij} \leq \sum_{(k,i) \in A} x_{ki} \qquad (i,j) \in A, \; i \neq 1. \tag{17}$$

This constraint states that, with the exception of the root vertex, only those vertices reached by an arc may have outgoing arcs. It is also useful to introduce other valid inequalities stating that each original edge can only be used in one direction:

$$x_{ij} + x_{ji} \leq 1 \qquad (i,j) \in E. \tag{18}$$

2.2 Miller-Tucker-Zemlin formulations

Different directed formulations can be obtained by using adaptations of the *Miller-Tucker-Zemlin* (MTZ) constraints (Miller et al., 1960) for the TSP. Gouveia (1995) has used this idea

to propose a formulation for the MST with hop constraints, while Khoury et al. (1993) have used the same set of constraints in a pure STP formulation. Later, Voß (1999) has adapted these formulations for the STP with hop constraints. The basic idea of the MTZ constraints is to introduce potential variables associated with each vertex of the graph and impose that in every path of the solution, the potential variables become larger when the distance from the root vertex increases. Let u_i be a real-valued potential variable associated with vertex i. Again, h is the maximum number of arcs between any vertex and the root vertex. One can model the Budget STPRBH as follows:

Aggregated Miller-Tucker-Zemlin (AMTZ) formulation

$$\text{Maximize} \sum_{i \in V} r_i y_i \tag{19}$$

subject to

$$\sum_{(j,i) \in A} x_{ji} = y_i, \qquad i \in V \backslash \{1\}, \tag{20}$$

$$\sum_{(i,j) \in A} c_{ij} x_{ij} \leq b, \tag{21}$$

$$u_1 = 0, \tag{22}$$

$$y_i \leq u_i \leq h y_i, \qquad i \in V \backslash \{1\}, \tag{23}$$

$$(h+1)x_{ij} + u_i - u_j + (h-1)x_{ji} \leq h \qquad (i,j) \in A, \tag{24}$$

$$y_1 = 1, \tag{25}$$

$$x_{ij} \in \{0,1\}, \qquad (i,j) \in A, \tag{26}$$

$$y_i \in \{0,1\}, \qquad i \in V \backslash \{1\}. \tag{27}$$

Note that connectivity constraints (11) have been excluded from this new formulation since the MTZ constraints (22)–(24) are sufficient to ensure that the solution contains no cycles. Indeed, constraints (24) guarantee that each vertex in a path has an associated potential variable u_i larger than that of its predecessors, which is impossible in a circuit. Constraints (22)–(23) guarantee that the largest potential in the solution does not exceed h.

Constraints (17) and (18) can again be used to strengthen the formulation. Besides these constraints, other valid inequalities can be added to the model. For example, one can say that for vertices with zero revenue, constraints (23) can be tightened, as

$$y_i \leq u_i \leq (h-1)y_i \qquad i \in V \backslash \{1\}, r_i = 0, \tag{28}$$

which are valid since there is always an optimal solution containing no non-profitable vertices as leaves. The upper bounds on these constraints can be further strengthened, as proposed by

Voß (1999), by considering the arcs originating at the root node:

$$y_i \leq u_i \leq (h-1)y_i - (h-2)x_{1i} \qquad i \in V\backslash\{1\}, r_i = 0. \tag{29}$$

The consideration of the arcs from the root enables the strengthening of constraints (23) also for the case of profitable vertices:

$$y_i \leq u_i \leq hy_i - (h-1)x_{1i} \qquad i \in V\backslash\{1\}, r_i > 0. \tag{30}$$

Finally, Voß also proposes several strengthening for onstraints (24), depending on the type of the vertices involved: mandatory vertices or Steiner vertices. We can adapt one of these modifications for the case where the two vertices involved in the constraint are non-profitable. In this case, we know that none of the vertices is a leaf and, therefore, the difference between their potentials is bounded by $h-1$. Constraints (24) then become:

$$hx_{ij} + u_i - u_j + (h-2)x_{ji} \leq h-1 \qquad (i,j) \in A, r_i = r_j = 0. \tag{31}$$

Besides strengthening the formulation with valid inequalities, Voß has proposed the introduction of three-index variables x_{ij}^p, equal to 1 if and only if arc (i,j) belongs to the solution, p hops away from the root. With this new set of variables and the coupling constraints

$$x_{ij} = \sum_{p=1}^{h} x_{ij}^p, \qquad (i,j) \in A, \tag{32}$$

valid inequalities (17) can be strengthened and rewritten as

$$x_{ij}^p \leq \sum_{(k,i)\in A} x_{ki}^{p-1} \qquad (i,j) \in A, \quad p = 2, \ldots, h. \tag{33}$$

The author has reported a beneficial effect of including this disaggregated version of the valid inequalities. Section 4 shows that these so-called *continuity constraints* are also effective for the STPRBH. The complete modified formulation is presented below:

Disaggregated Miller-Tucker-Zemlin (DMTZ) formulation

$$\text{Maximize} \sum_{i\in V} r_i y_i \tag{34}$$

subject to

$$\sum_{(j,i)\in A} x_{ji} = y_i, \qquad i \in V\backslash\{1\}, \tag{35}$$

$$\sum_{(i,j)\in A} c_{ij}x_{ij} \leq b, \tag{36}$$

$$y_i \leq u_i \leq hy_i, \qquad i \in V\backslash\{1\}, \tag{37}$$

$$(h+1)x_{ij} + u_i - u_j + (h-1)x_{ji} \leq h, \qquad (i,j) \in A, \tag{38}$$

$$x_{ij}^p \leq \sum_{(k,i)\in A} x_{ki}^{p-1}, \qquad p = 2,\dots h, (i,j) \in A, i \neq 1, \tag{39}$$

$$x_{ij} = \sum_{p=1}^{h} x_{ij}^p, \qquad (i,j) \in A, \tag{40}$$

$$u_j = \sum_{p=1}^{h} \sum_{(i,j)\in A} p x_{ij}^p, \qquad j \in V\backslash\{1\}, \tag{41}$$

$$x_{ij}^1 = 0, \qquad (i,j) \in A, i \neq 1, \tag{42}$$

$$x_{1j}^p = 0, \qquad j \in V\backslash\{1\}, p = 2,\dots,h, \tag{43}$$

$$u_1 = 0, \tag{44}$$

$$y_1 = 1, \tag{45}$$

$$y_i, x_{ij} \in \{0,1\}, \qquad i \in V\backslash\{1\}, (i,j) \in A. \tag{46}$$

Constraints (41) link the three-index variables with the potential variables, while constraints (42)–(43) indicate that hop count starts at the root vertex.

Note that the liftings (29)–(31) can also be used in the disaggregated formulation. In the presence of 3-index variables x_{ij}^p, we propose four new families of valid inequalities:

$$\sum_{p=1}^{h} p x_{ij}^p \geq u_i - h + (h+1)x_{ij}, \qquad (i,j) \in A, \tag{47}$$

$$x_{ij}^p \geq \sum_{(k,i)\in A} x_{ki}^{p-1} + x_{ij} - 1, \qquad (i,j) \in A, i \neq 1, p = 2,\dots,h, \tag{48}$$

$$\sum_{(i,j)\in A} x_{ij}^p \geq \sum_{(k,i)\in A} x_{ki}^{p-1}, \qquad (i,j) \in A, i \neq 1, r_i = 0, \tag{49}$$

$$x_{ij}^p \leq 1 - \sum_{(k,i)\in A} \sum_{t=p}^{h} x_{ki}^t, \qquad (i,j) \in A, i \neq 1, p = 2,\dots,h. \tag{50}$$

Inequalities (47) are a linearization of

$$\sum_{p=1}^{h} p x_{ij}^p \geq (u_i + 1)x_{ij}, \qquad (i,j) \in A, \tag{51}$$

which indicate that the arcs leaving a certain vertex are the $(u_i+1)^{th}$ hops from the root. Inequalities (48) in turn, impose that if arc (i,j) is in the solution it is h_i+1 hops away from the root vertex, where h_i is the hop count of the arc that enters the origin vertex i. Inequalities (49) state that no non-profitable vertex is a leaf in the solution. These are a three-index version of the also valid inequalities

$$\sum_{(i,j)\in A} x_{ij} \geq \sum_{(k,i)\in A} x_{ki}, \qquad (i,j) \in A, r_i = 0, i \neq 1. \tag{52}$$

Finally, inequalities (50) state that an arc cannot be p hops away from the root if its origin vertex has any incoming arc that is more than p hops away from the root. In Section 4 we test these sets of valid inequalities and retain only those which have a positive effect.

3 Branch-and-cut Algorithms

In Section 2 we have presented four mathematical formulations for the STPRBH. Two of these formulations use an exponential number of constraints based on subtour elimination constraints (UDFJ and DDFJ), while two others use variations of MTZ constraints (AMTZ and DMTZ). Because of the large number of constraints, it is impractical to use commercial solvers for formulations UDFJ and DDFJ, even for mid-size instances. We propose a branch-and-cut algorithm containing two separation procedures which generate violated constraints of type (12) and (13) only when these are found to be violated during the branch-and-cut process. We concentrate on the directed DDJF formulation since it has proved to be stronger than undirected formulations (see, e.g., Ljubić et al., 2005). Section 3.1 explains the two separation procedures and the overall branch-and-cut algorithms.

The number of constraints in formulations AMTZ and DMTZ is polynomial and, therefore, one can make use of commercial LP solvers for the problem as long as the instance size is not too large. However, in the case of the DMTZ formulation, it has proved beneficial to relax constraints (39) and (41) and add them as cuts in the model when they are found to be violated during the branching process. Section 3.2 presents this second branch-and-cut approach.

Finally, in Section 3.3, we describe minor modifications to the algorithms and define the complete set of algorithms to test.

3.1 Branch-and-cut for formulation DDJF

We consider the model DDJF. The idea of our branch-and-cut algorithm is to initially relax connectivity constraints (11) and hop constraints (13) and add them as cuts only when they become violated. It is therefore necessary to develop appropriate separation procedures.

The first separation procedure takes care of the connectivity of the solution. This procedure is adapted from the work of Ljubić et al. (2005). It consists of identifying disconnected vertices by means of a maximum flow algorithm and of adding the associated violated constraints of type (11). The second separation procedure identifies violated hop constraints of type (13) by finding paths from the root to any vertex containing more arcs than the allowed hop limit. These two routines are now presented in detail.

3.1.1 Connectivity separation procedure

The connectivity separation procedure exploits the fact that constraints (11) imply the connectivity of all selected vertices in the graph, in particular, the connectivity of the root to all other

selected vertices. Indeed, given a partition $\{S, V\backslash S\}$ of V and a cut $C = \{(i,j) : i \in S, j \in V\backslash S\}$ in which the root and a given vertex k are on different sides of the cut, i.e., $1 \in S$ and $k \in V\backslash S$, let X_C be the sum of the x variables belonging to the cut: $X_C = \sum_{(i,j)\in C} x_{ij}$. The connectivity constraints guarantee that $X_C \geq y_k$. Now, if one uses the current x solution at a given node of the branching tree as arc capacities, it is possible to use a maximum flow algorithm to find the minimum cut, in terms of X_C, between the root and every other vertex k of the graph. If X_C is smaller than y_k, then a violated connectivity constraint has been identified and can be added to the problem.

In order to identify more violated inequalities at each iteration, Ljubić et al. (2005) have proposed the use of *nested cuts* and *back-cuts*. The nested cuts were introduced by Koch and Martin (1998). Once a violated cut is found, these cuts consist of setting to one the values of the x variables in the cut and of solving the minimum cut problem again, in the hope of finding another cut that does not include all of the previous arcs. The process is repeated until no more cuts can be identified or an upper limit is reached. The back-cuts were introduced by Chopra and Rao (1994). The idea is simply to reverse the flows and look for a minimum cut with i as origin vertex and the root vertex as destination. In our implementation, as suggested by Ljubić et al. (2005), the nested cuts and the back-cuts are combined, maximizing the number of violated constraints found at each iteration.

3.1.2 Hop constraints separation procedure

The search for violated hop constraints consists of exploring the current tree for paths containing more arcs than the hop limit. We concentrate on paths rooted at vertex 1 and we explore the subgraph induced by the positive y_i and x_{ij} variables. For a path $P = (i_1 = 1, \ldots, i_\ell)$, define $X_P = \sum_{t=2}^{\ell} x_{i_{t-1},i_t}$ as the sum of the x_{ij} variables in the path. The separation procedure gradually extends a path P rooted at 1 until the number of arcs in the path reaches $h+1$ and the sum of associated variables exceeds h, or this sum is less than the number of arcs minus one. In other words, these conditions are:

1) $\ell = h + 2$ and $X_P > h$,

2) $X_P < \ell - 2$.

In case 1) a violated hop constraint has been found and the cut $X_P \leq h$ is added to the model. In case 2) it is useless to continue exploring the branch, since no violated inequalities will be found. The complete separation procedure is described in Algorithm 1, where $f[i] = 1$ if vertex i can still been visited in the search ($i = 1, \ldots, n$), and 0 otherwise.

The idea of Algorithm 1 is to sequentially visit the connected vertices in the induced subgraph until one of conditions 1) or 2) is met. Again, if condition 1) is satisfied, a violated cut has been identified and is added to the model. In both cases, the algorithm blocks the access to the last vertex in the path and returns to the previous vertex, where the search is resumed. The

procedure continues until the algorithms finds itself in the root vertex, with no other vertices to visit.

Algorithm 1 Hop separation procedure

$\ell = 1$; $P = (i_\ell = 1)$; $X_P = 0$; $f = [0, 1, 1, \ldots 1]$;
loop
  if there exists $(i_\ell, j) \in A$, with $f[j] = 1$ **then**
    $\ell = \ell + 1$;
    $i_\ell = j$;
    $f[i_\ell] = 0$;
    $X_P = X_P + x_{i_{\ell-1}, i_\ell}$;
    if $\ell = h + 2$ and $X_P > h$ **then**
      Add the cut $X_P \leq h$;
      $f[k] = 1$, for all k such that $(i_\ell, k) \in A, k \neq i_{\ell-1}$;
      $X_P = X_P - x_{i_{\ell-1}, i_\ell}$;
      $\ell = \ell - 1$;
    else
      if $X_P < \ell - 2$ **then**
        $X_P = X_P - x_{i_{\ell-1}, i_\ell}$;
        $\ell = \ell - 1$;
      end if
    end if
  else
    if $\ell = 1$ **then**
      Stop.
    else
      $X_P = X_P - x_{i_{\ell-1}, i_\ell}$;
      $f[k] = 1$, for all k such that $(i_\ell, k) \in A, k \neq i_{\ell-1}$;
      $\ell = \ell - 1$;
    end if
  end if
end loop

3.1.3 Complete description of the branch-and-cut algorithm

The branch-and-cut algorithm starts by solving formulation (9)–(16) with constraints (11) and (13) relaxed. At the root node, valid inequalities (17) and (18) are also added to the model. At each node of the branch-and-cut tree, the algorithm looks for violated connectivity and hop constraints. If any is found, the linear problem at the node is solved again and the algorithm looks for new valid inequalities. Algorithm 2 presents an overview of the procedure.

Algorithm 2 Branch-and-cut method (DDFJ)

```
while constraints added do
  Solve root node.
  Look for violated connectivity constraints (11).
  Look for violated hop constraints (13).
end while
if solution is integer then
  END.
else
  Start branching
  while there exist open nodes do
    Branch or change current node
    while constraints added do
      Solve current node.
      Look for violated connectivity constraints (11).
      Look for violated hop constraints (13).
    end while
  end while
end if
```

3.2 Branch-and-cut for formulation DMTZ

Although formulation DMTZ contains a polynomial number of constraints, we have decided to test the efficacy of a branch-and-cut algorithm that relaxes a fraction of these constraints and inserts them into the model only when they are found to be violated. We have relaxed constraints (39) and (41) associated with the three-index variables.

We use a simple separation procedure to check whether each one of the relaxed constraints is violated. If this is the case, the constraint is added in the formulation and remains in it until the end of the branch-and-cut process. Algorithm 3 presents an overview of this procedure.

Algorithm 3 Branch-and-cut method (DMTZ)

```
while constraints added do
  Solve root node.
  Look for violated constraints (39) and (41).
end while
if solution is integer then
  END.
else
  Start branching
  while there exists open nodes do
    Branch or change current node.
    while constraints added do
      Solve current node.
      Look for violated constraints (39) and (41).
    end while
  end while
end if
```

3.3 Other branch-and-cut algorithms

In Sections 3.1 and 3.2 we have introduced four algorithms for the resolution of the STPRBH. A simple algorithmic approach is to use a commercial mixed-integer package to solve formulations AMTZ (Algorithm A1) and DMTZ (Algorithm A2). We also have the possibility of using the two proposed branch-and-cut algorithms that relax either some of the constraints containing three-index variables on formulation DMTZ (Section 3.2) or the connectivity and hop constraints on formulation DDJF (Section 3.1). There algorithms are referred to as A3 and A4, respectively.

Three other algorithms, called A5, A6 and A7, can be constructed by using the connectivity and hop constraints as cuts when solving the STPRBH via the AMTZ or DMTZ formulations: during the branching, violated constraints (11) and (13) are added to the model. Note that for the DMTZ formulation this gives rise to two new possibilities since the model can originally be solved with or without the relaxation of the three-index constraints. Table 1 presents a summary of the algorithms.

4 Computational Experiments

The branch-and-cut algorithms just described were implemented within the CPLEX 9.1.3 framework and run on an AMD Opteron machine with a 2390MHz CPU and 16Gb RAM, under Linux.

Table 1: Summary of the algorithms

Algorithm	Formulation at root node	Cuts
A1	AMTZ	
A2	DMTZ	
A3	DMTZ \{(39), (41)}	(39), (41)
A4	DDFJ \{(11), (13)}	(11), (13)
A5	AMTZ	(11), (13)
A6	DMTZ	(11), (13)
A7	DMTZ \{(39), (41)}	(11), (13), (39), (41)

4.1 Test instances

We have tested the proposed valid inequalities and branch-and-cut algorithms on two set of instances obtained from the OR-Library (Beasley, 1990). We have adapted these instances for the STPRBH, by using the mandatory vertices as profitable vertices with revenue generated randomly by a discrete uniform distribution over the interval $[1, 100]$. The Steiner vertices were attributed a zero revenue. For each instance of the first group, 12 scenarios were analyzed, obtained from four values for the hop limit, $h = 3, 6, 9$ and 12, and three values for the budget, $b = s, s/5$ and $s/10$, where $s = \sum_{(i,j)\in E} c_{ij}$. For the second group of instances, we used $h = 5, 15$ and 25 and $b = s, s/10$ and $s/30$. Since vertex 1 is not always a mandatory vertex in the original instances, the root vertex was chosen to be the mandatory vertex with the smallest index. The instance sizes range from 50 vertices and 63 arcs to 500 vertices and 625 arcs.

It is interesting to observe how the budget and hop constraints interact, making the solutions of different scenarios for the same instance completely different. Instance $MSteinb8$ is a good example and is detailed in Table 2 which shows, for each scenario, the optimal solution value, the number of non-zero revenue vertices reached, and the percentage of the total revenues collected. The table is linked to Figures 1 and 2 which depict the solutions. The values inside the vertices are the vertex numbers, while the values next to the arcs are their cost. A vertex is represented by a circle if its revenue is zero and by a square, otherwise.

For the first three scenarios, there is no budgetary limit and, therefore, the hop constraint is the only restriction preventing the solution from reaching all profitable vertices. For $h = 3$, only two vertices can be reached, yielding a total revenue of 85. For $h = 6$, the total revenue increases to 581 (given by seven profitable vertices) but it is only for $h = 9$ and $h = 12$ that all 19 profitable vertices are reached. A similar analysis can be made for a constant value of the hop limit, for example, $h = 9$: in this case, the reduction in the budget reduces the total collectable revenue from 836 ($b = s$) to 761 ($b = s/5$) and finally to 537 ($b = s/10$). In the sequel, we describe the results for all the instances. We first run some preliminary tests to determine the impact of the valid inequalities, and we then compare the performance of the different algorithms presented in Section 3.3.

Scenario	b	h	Optimal solution value	Profitable vertices reached	% Revenue collected	Figure
1	s	3	85	2	10	1(a)
2	s	6	581	13	70	1(b)
3	s	9	836	19	100	1(c)
4	s	12	836	19	100	1(c)
5	s/5	3	85	2	10	1(a)
6	s/5	6	535	11	64	1(d)
7	s/5	9	761	16	91	1(e)
8	s/5	12	832	18	99	2(a)
9	s/10	3	85	2	10	1(a)
10	s/10	6	346	6	41	2(b)
11	s/10	9	537	10	64	2(c)
12	s/10	12	537	10	64	2(c)

Table 2: Optimal solutions for instance *MSteinb8* for the 12 different scenarios

4.2 Preliminary tests

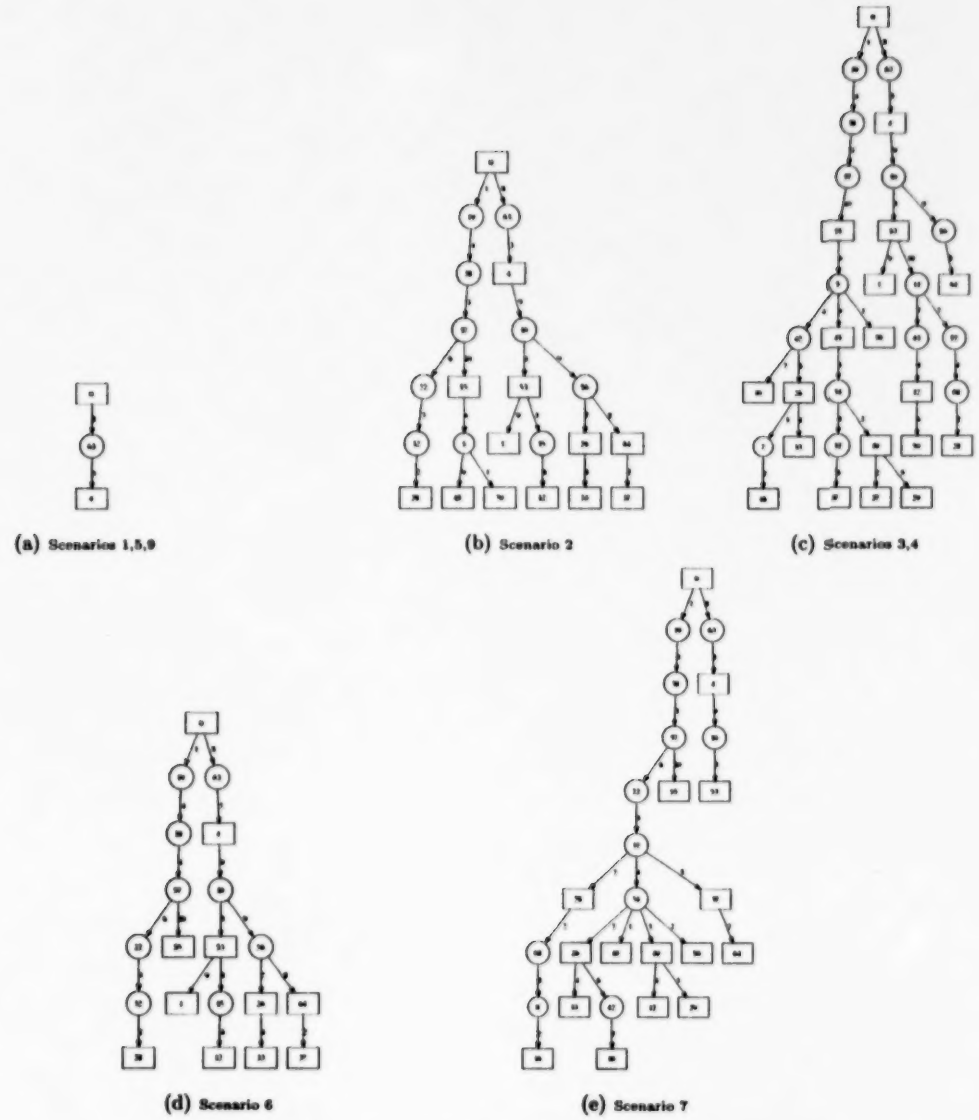
We have presented in Section 3.3 several algorithms for the STPRBH based on the formulations and separation procedures developed earlier. One can run each of these algorithms with or without the associated valid inequalities. Due to the large number of possible combinations of algorithms that could be used, we have decided to run some preliminary tests on a limited number of instances. In these tests, some of the possibilities presented in Table 1 were dominated by others. Indeed, the two algorithms based on the AMTZ formulation ($A1$ and $A5$) were always dominated by the associated disaggregated version. Moreover, algorithm $A6$ was dominated by $A2$.

Concerning the valid inequalities, the tests indicate that valid inequalities (17) and (18) proved to be always helpful and, since they are valid for all the formulations, they were added in all tests. For the AMTZ formulation, the four valid inequalities (28) to (31) were available. The tests showed that there was no clear advantage in using any of these inequalities and they were therefore discarded. The same happened for inequalities (47)–(50), valid for the DMTZ formulation, whose inclusion did not improve performance. We also tried to introduce these inequalities as cuts when they became violated, but this was not helpful either.

Table 3 shows the final configurations that were tested in the complete test presented in the next section.

Algorithm	Formulation at the root node	Cuts
A2	DMTZ +{(17), (18)}	
A3	DMTZ +{(17), (18)}\{(39), (41)}	(39), (41)
A4	DDFJ +{(17), (18)}\{(11), (13)}	(11), (13)
A7	DMTZ +{(17), (18)}\{(39), (41)}	(11), (13), (39), (41)

Table 3: Non-dominated algorithms



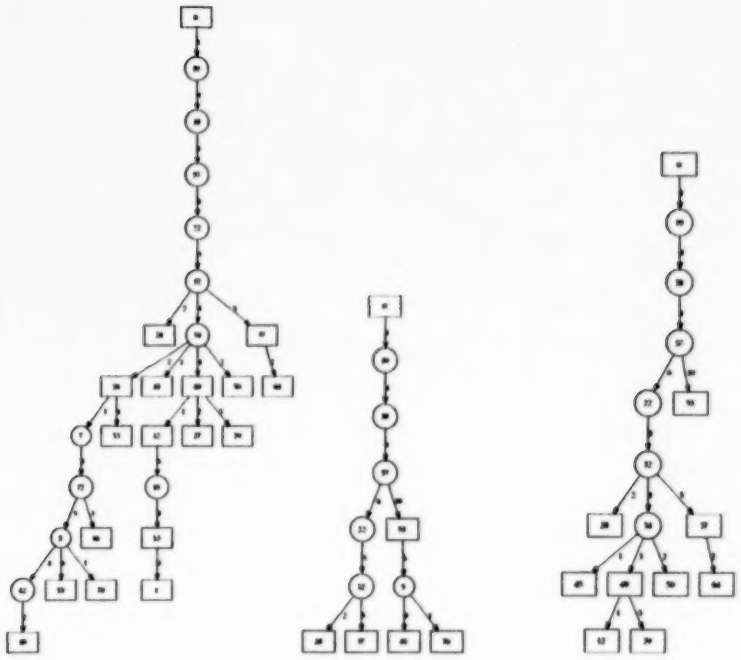
(a) Scenarios 1,5,9 (b) Scenario 2 (c) Scenarios 3,4

(d) Scenario 6 (e) Scenario 7

Figure 1: Solutions for instance *MSteinb8* in the different scenarios.

4.3 Complete tests

The four algorithms presented in Table 3 were tested on the two sets of instances described in Section 4.1. A maximum CPU time of two hours was allowed for the solution of each scenario of each instance. For some scenarios this was not enough for some algorithms to converge. In Tables 5 to 11 we present the detailed results of our computational tests. Besides the time needed by the algorithm, we also report the number of cuts (11) and (13) for algorithms *A4* and *A7* and the combined number of cuts (39) and (41) for algorithm *A7*. In this case, we also



(a) Scenario 8 (b) Scenario 10 (c) Scenarios 11,12

Figure 2: Solutions for instance *MSteinb8* in the different scenarios (continued).

indicate the percentage of the total number of constraints (39) and (41) that were added to the model. For each algorithm, the tables also show the number of branch-and-cut nodes explored. In these tables, when the time limit is not enough for the algorithm to reach an optimal solution, we present the MIP gap in brackets. The entry "[inf]" indicates that no feasible solution was identified within the time limit.

For the scenarios with a small hop limit h the MTZ formulations are clearly superior. Indeed, $A2$ wins in 50 out of 54 cases for instances *Msteinb* with $h = 3$, and in all instances *Steinc* with $h = 5$. The reason for this behavior seems to be the fact that formulation $A2$ is extremely strong and its size is reasonable for small values of h. This hypothesis can be confirmed by the fact that a great part of these instances are solved in the root node. For small values of h, $A3$ is also effective but less than $A2$. However, formulation $A2$ quickly grows with h and, for this reason, the relaxation of constraints (39) and (41) starts to pay off, as one can see in a direct comparison of algorithms $A2$ and $A3$ for the *MSteinb* instances with $h = 12$, for example.

Interestingly, Algorithm $A4$ behaves in a complete different manner. Indeed, for small values of h, the number of generated constraints (11) and (13) is huge, making the algorithm very inefficient. As soon as h grows, however, fewer cuts are needed for convergence and $A4$ clearly becomes the most efficient approach (see, for example, the results for instances *MSteinb* for $h = 12$ in Table 8). As expected, algorithm $A7$ lies somewhere between algorithms $A2$ and $A4$.

Group	h	A2		A3		A4		A7	
		Best	Unsolved	Best	Unsolved	Best	Unsolved	Best	Unsolved
MStein	3	50	0	0	0	0	25	41	0
MStein	6	44	0	1	0	5	20	18	0
MStein	9	5	0	10	0	36	4	4	0
MStein	12	0	0	5	0	49	0	0	0
SteinC	5	30	0	0	0	0	30	0	0
SteinC	15	10	4	7	9	11	12	2	3
SteinC	25	0	24	2	9	23	6	0	18
Total		139	28	25	18	124	97	65	21

Table 4: Summary of results

Table 4 summarizes the results of these tests. For the four possible solution methods and the two groups of instances, we indicate the number of scenarios for which the method was the most efficient and the number of scenarios the method could not solve within the imposed time limit. One can observe again the clear polarity between methods $A2$ and $A4$. Indeed, for instances with a small h, $A2$ seems to be the most efficient method while for instances with a large h, $A4$ yields better results. As indicated earlier, this is somehow expected since for increasing values of h two phenomena happen: on the one hand, the size of formulation DMTZ quickly increases (especially the number of variables) and, on the other hand, the number of cuts necessary to find a feasible solution in algorithm $A4$ becomes smaller. Concerning algorithms $A3$ and $A7$, although they are never the *most efficient* for a given group of instances, they prove to be the more robust methods, not being able to solve only 18 and 21 out of the 306 instances, respectively.

5 Conclusions

We have proposed several formulations for a modified version of the Steiner Tree Problem with revenues, including budget and hop constraints. These formulations were used to develop branch-and-cut algorithms for the problem. Computational tests have shown that the proposed algorithms are able to solve a majority of scenarios for instances with up to 500 vertices and 625 arcs. Moreover, our results indicate that the best algorithm clearly depends on the number of allowed hops. For small hop values, algorithms based on Miller-Tucker-Zemlin formulations are clearly superior, while for large hop values the algorithm based on Dantzig-Fulkerson-Johnson formulation is the most efficient.

Acknowledgments

This work was supported by the Brazilian Federal Agency for Post-Graduate Education - CAPES, under grant BEX 1097/01-6, and by the Canadian Natural Science and Engineering Research Council under grants 227837-04 and OGP0039682. This support is gratefully acknowledged. The authors thank Stefan Ropke for kindly supplying his maxflow algorithm implementation.

Instance					A2		A3			A4				A7			
Name	n	$\|E\|$	b	Opt	B&B nodes	Seconds	B&B nodes	Cuts (%)	Seconds	B&B nodes	Sep	Hops	Seconds	B&B nodes	Sep	Hops	Seconds
Mstеinb1	50	63	1	140	0	0.01	0	5 (1.68 %)	0.02	440	116	272	0.5	0	0	0	0.01
			5	140	0	0.01	2	9 (3.02 %)	0.04	394	96	150	0.37	0	0	0	0.01
			10	140	0	0.01	0	15 (5.03 %)	0.03	72	26	47	0.14	0	0	0	0.01
Mstеinb2	50	63	1	182	0	0.01	0	44 (14.77 %)	0.04	608	216	90	0.43	0	0	0	0.01
			5	182	0	0.01	0	64 (21.48 %)	0.06	258	94	47	0.25	0	0	0	0.01
			10	182	0	0.01	0	43 (14.43 %)	0.04	194	74	41	0.25	0	0	0	0.01
Mstеinb3	50	63	1	253	0	0.01	0	89 (29.87 %)	0.05	190	144	7	0.25	0	0	0	0.01
			5	253	0	0.01	0	79 (26.51 %)	0.07	105	96	6	0.19	0	0	0	0.01
			10	253	0	0.01	0	68 (22.82 %)	0.07	87	66	4	0.16	0	0	0	0.01
Mstеinb4	50	100	1	341	0	0.01	0	54 (18.12 %)	0.13	54750	6846	24736	[inf]	12	2	0	0.01
			5	341	0	0.01	0	52 (17.45 %)	0.18	64464	7810	37499	[inf]	8	2	0	0.01
			10	341	0	0.01	0	44 (14.77 %)	0.14	17485	2396	24740	296.45	0	0	0	0.01
Mstеinb5	50	100	1	588	0	0.01	9	82 (18.39 %)	0.17	110767	9916	27292	[inf]	0	0	0	0.01
			5	588	0	0.01	0	96 (21.52 %)	0.24	76690	6870	47978	[inf]	0	0	0	0.02
			10	565	0	0.01	0	51 (11.43 %)	0.07	27885	3154	30296	425.22	0	0	0	0.01
Mstеinb6	50	100	1	1040	0	0.02	16	180 (40.36 %)	0.28	346809	19700	48115	[inf]	0	0	0	0.02
			5	1035	0	0.02	17	170 (38.12 %)	0.34	224830	21005	39790	[64%]	0	0	0	0.02
			10	610	41	0.06	58	124 (27.80 %)	0.64	32100	4498	24208	859.75	16	2	40	0.09
Mstеinb7	75	94	1	87	0	0.01	0	3 (0.67 %)	0.04	85	56	3	0.16	0	0	0	0.01
			5	87	0	0.02	0	3 (0.67 %)	0.05	63	50	11	0.19	0	0	0	0.01
			10	87	0	0.01	0	3 (0.67 %)	0.04	110	64	1	0.14	0	0	0	0.01
Mstеinb8	75	94	1	85	0	0.01	0	75 (16.78 %)	0.07	119	78	3	0.21	0	0	0	0.01
			5	85	0	0.01	0	72 (16.11 %)	0.09	122	92	3	0.33	0	0	0	0.01
			10	85	0	0.01	0	51 (11.41 %)	0.08	61	62	3	0.18	0	0	0	0.01
Mstеinb9	75	94	1	596	0	0.01	0	149 (33.33 %)	0.09	52342	7698	3458	141.98	0	0	0	0.02
			5	596	0	0.01	0	126 (28.19 %)	0.15	26481	6020	22166	700.93	0	0	0	0.02
			10	483	3	0.02	16	112 (25.06 %)	0.2	1497	450	674	3.08	0	0	3	0.02
Mstеinb10	75	150	1	319	0	0.02	0	123 (18.33 %)	0.3	130267	12174	21183	[inf]	0	0	0	0.02
			5	319	0	0.02	42	97 (14.46 %)	0.6	108525	7044	16258	[inf]	0	0	0	0.02
			10	319	0	0.02	0	119 (17.73 %)	0.4	124916	10880	38385	[71%]	0	0	0	0.02
Mstеinb11	75	150	1	316	0	0.02	25	185 (27.57 %)	0.51	71706	10152	12909	[inf]	0	0	0	0.02
			5	316	0	0.02	81	223 (33.23 %)	1.16	85095	12944	19137	[inf]	0	0	0	0.02
			10	305	0	0.02	2	163 (24.29 %)	0.38	56827	10420	34909	[79%]	0	0	0	0.02
Mstеinb12	75	150	1	1169	0	0.02	43	340 (50.67 %)	0.82	480074	27260	16502	[inf]	0	0	0	0.03
			5	1169	0	0.02	0	284 (42.32 %)	0.41	385334	26928	21709	[inf]	0	0	0	0.03
			10	1017	3	0.03	94	205 (30.55 %)	1.38	233496	21268	16518	[46%]	0	0	3	0.05
Mstеinb13	100	125	1	147	0	0.02	0	32 (5.37 %)	0.07	1670	430	149	3.1	0	0	0	0.01
			5	147	0	0.02	0	47 (7.89 %)	0.11	5186	1524	4517	55.48	0	0	0	0.01
			10	147	0	0.01	0	47 (7.89 %)	0.09	1427	358	268	2.74	0	0	0	0.02
Mstеinb14	100	125	1	263	0	0.02	0	12 (2.01 %)	0.06	789	272	45	1.98	0	0	0	0.01
			5	263	0	0.01	0	26 (4.36 %)	0.09	6110	2204	15724	268.13	0	0	0	0.02
			10	263	0	0.01	0	25 (4.19 %)	0.09	915	396	158	3.31	0	0	0	0.02
Mstеinb15	100	125	1	1061	0	0.02	0	127 (21.31 %)	0.11	333425	37488	13533	[58%]	0	0	0	0.02
			5	1061	0	0.02	0	128 (21.48 %)	0.16	110851	21758	28402	[31%]	0	0	0	0.02
			10	830	0	0.02	3	93 (15.60 %)	0.18	4999	1656	4145	48.07	0	0	0	0.03
Mstеinb16	100	200	1	479	0	0.03	0	131 (14.62 %)	0.37	109429	10562	8213	[inf]	0	0	0	0.03
			5	479	0	0.03	0	123 (13.73 %)	0.5	76921	7458	11080	[inf]	0	0	0	0.03
			10	479	0	0.03	0	113 (12.61 %)	0.55	130863	13476	20366	[inf]	0	0	0	0.03
Mstеinb17	100	200	1	254	0	0.02	0	193 (21.54 %)	0.37	48738	9112	5154	[80%]	0	0	0	0.02
			5	254	0	0.02	0	208 (23.21 %)	0.46	41440	7310	8983	[84%]	0	0	0	0.02
			10	254	0	0.02	0	181 (20.20 %)	0.54	10861	2062	3400	78.73	0	0	0	0.03
Mstеinb18	100	200	1	1298	0	0.03	0	423 (47.21 %)	0.5	455560	19216	11440	[inf]	0	0	0	0.03
			5	1298	0	0.03	0	393 (43.86 %)	0.74	334184	20845	7723	[inf]	0	0	0	0.03
			10	1132	3	0.05	35	349 (38.95 %)	1.31	531476	31841	8310	[76%]	0	0	3	0.08

Table 5: Results - Instances MStein, $h = 3$

Instance					A2		A3			A4				A7			
Name	n	$\|E\|$	h	Opt	B&B nodes	Seconds	B&B nodes	Cuts (%)	Seconds	B&B nodes	Sep	Hops	Seconds	B&B nodes	Sep	Hops	Seconds
Msteinb1	50	63	1	403	0	0.02	52	200 (29.85 %)	0.3	744	162	1395	2.09	0	0	0	0.02
			5	403	0	0.03	7	101 (15.07 %)	0.18	129	56	173	0.4	0	0	0	0.03
			10	341	0	0.03	0	41 (6.12 %)	0.03	0	0	0	0.01	0	0	0	0.03
Msteinb2	50	63	1	612	0	0.01	72	239 (35.67 %)	0.6	1647	468	5300	10.02	0	0	0	0.02
			5	491	3	0.05	83	318 (47.46 %)	1.03	351	172	455	1.02	0	0	3	0.05
			10	300	5	0.04	74	79 (11.79 %)	0.39	11	34	45	0.17	0	0	5	0.05
Msteinb3	50	63	1	963	0	0.01	73	402 (60.00 %)	1.36	4218	718	5337	11.85	0	0	0	0.02
			5	807	2	0.03	24	326 (48.66 %)	0.57	398	122	429	0.91	0	0	2	0.04
			10	624	0	0.02	61	59 (8.81 %)	0.32	44	44	12	0.1	0	0	0	0.02
Msteinb4	50	100	1	455	3	0.83	264	470 (70.15 %)	4.24	6330	346	414	3.06	311	12	5	1.59
			5	455	0	0.32	77	247 (36.87 %)	1.38	80962	5046	51151	[inf]	220	4	0	0.36
			10	447	54	1.97	351	277 (41.34 %)	3.73	3392	220	454	3.68	396	36	40	2.31
Msteinb5	50	100	1	666	1	0.34	237	601 (57.79 %)	7.01	4006	184	107	1.21	7	2	0	0.23
			5	666	2	0.49	28	109 (10.48 %)	0.41	3	0	0	0.03	7	2	0	0.22
			10	643	20	0.55	181	139 (13.37 %)	0.9	408	28	31	0.23	0	0	20	0.58
Msteinb6	50	100	1	1269	23	3.48	108	239 (22.98 %)	0.91	532223	16224	81369	[inf]	10	6	12	2.9
			5	1257	41	1.79	368	535 (51.44 %)	6.93	87219	5936	44656	3085.22	191	24	57	2.18
			10	839	21	1.01	277	328 (31.54 %)	2.84	1408	222	1091	4.55	316	18	48	1.73
Msteinb7	75	94	1	627	0	0.02	155	503 (50.05 %)	3.9	5777	1468	2941	27.51	0	0	0	0.02
			5	627	0	0.02	75	434 (43.18 %)	1.3	209	102	98	0.67	0	0	0	0.02
			10	432	2	0.05	114	56 (5.57 %)	0.68	47	58	67	0.34	0	0	2	0.05
Msteinb8	75	94	1	581	0	0.02	163	546 (54.33 %)	4.01	21152	4722	24770	517.98	0	0	0	0.02
			5	535	0	0.02	40	520 (51.74 %)	1.2	10899	2498	13851	148.23	0	0	0	0.03
			10	346	0	0.03	134	473 (47.06 %)	2.81	952	320	258	1.64	0	0	0	0.03
Msteinb9	75	94	1	1698	0	0.04	96	304 (30.25 %)	0.55	134894	15024	100630	[13%]	0	0	0	0.05
			5	1343	11	0.24	28	287 (28.56 %)	0.9	1247	360	764	3.55	0	0	11	0.28
			10	816	35	0.36	118	175 (17.41 %)	0.83	47	26	35	0.2	0	0	35	0.41
Msteinb10	75	150	1	702	11	3	304	918 (58.66 %)	21	262159	9428	12296	[inf]	128	12	6	2.72
			5	702	10	2.83	189	626 (40.00 %)	11.07	39042	2186	8397	531.61	55	8	2	1.01
			10	668	89	3.74	484	378 (24.15 %)	6.23	29087	1330	2930	186.9	652	24	148	4.62
Msteinb11	75	150	1	893	20	2.59	341	853 (54.50 %)	12.74	257468	7650	44915	[inf]	16	2	20	3.76
			5	893	8	2.16	585	838 (53.55 %)	18.46	199396	11208	28211	[inf]	121	4	2	1.56
			10	829	6	0.9	553	553 (35.34 %)	11.83	13395	656	1015	21	898	36	2	0.94
Msteinb12	75	150	1	1867	5	1.52	210	585 (37.38 %)	2.85	543501	17946	40729	[inf]	4	4	9	1.96
			5	1847	46	3.53	584	719 (45.94 %)	14.35	824712	29864	17476	[8%]	148	14	73	4.56
			10	1384	2	0.23	54	226 (14.44 %)	1.76	2642	228	300	3.31	0	0	2	0.25
Msteinb13	100	125	1	785	0	0.03	86	569 (42.46 %)	3.25	63447	11326	50946	[inf]	0	0	0	0.04
			5	674	42	0.21	80	583 (43.51 %)	2.65	5847	1890	13399	186.06	0	0	42	0.32
			10	452	18	0.18	90	227 (16.94 %)	1.89	92	40	65	0.54	0	2	16	0.24
Msteinb14	100	125	1	1296	0	0.03	94	702 (52.39 %)	3.24	51486	7774	65461	[inf]	0	0	0	0.03
			5	977	31	0.11	388	528 (39.40 %)	5.98	2662	1126	1103	18.04	2	2	43	0.23
			10	595	1	0.07	17	44 (3.28 %)	0.15	4	8	6	0.07	2	2	1	0.08
Msteinb15	100	125	1	2555	0	0.06	99	606 (45.22 %)	1.93	281664	24290	49401	[inf]	0	0	0	0.08
			5	1858	6	0.19	224	387 (28.88 %)	2.88	116	30	159	0.95	0	0	6	0.23
			10	1040	45	0.29	265	234 (17.46 %)	2.83	140	82	172	0.98	0	2	45	0.39
Msteinb16	100	200	1	840	0	0.23	389	721 (34.50 %)	12.17	236990	7584	17892	[inf]	0	0	0	0.23
			5	840	0	0.23	376	850 (40.67 %)	20.94	165222	6766	9697	[inf]	0	0	0	0.23
			10	767	106	4.32	628	918 (43.92 %)	25.89	118739	7866	11418	[92%]	164	24	117	6.35
Msteinb17	100	200	1	1299	0	0.1	484	1447 (69.23 %)	109.82	142261	8412	5041	[inf]	8	2	0	0.11
			5	1299	2	0.23	688	1359 (65.02 %)	177.01	347899	13742	8321	[inf]	11	2	0	0.18
			10	1091	0	0.09	467	1078 (51.58 %)	28.63	196087	16932	23666	[inf]	0	0	0	0.09
Msteinb18	100	200	1	2585	5	4.05	504	1058 (50.62 %)	17.76	1828387	33172	15558	[inf]	22	2	7	5.51
			5	2575	17	2.98	233	805 (38.52 %)	12.4	684522	17173	10929	[4%]	120	8	37	3.11
			10	1917	0	0.38	218	554 (26.51 %)	11.57	588941	25601	19713	[23%]	17	2	0	0.4

Table 6: Results - Instances MStein, $h = 6$

Instance					A2		A3			A4				A7			
Name	n	$\|E\|$	h	Opt	B&B nodes	Seconds	B&B nodes	Cuts (%)	Seconds	B&B nodes	Sep	Hops	Seconds	B&B nodes	Sep	Hops	Seconds
Msteinb1	50	63	1	467	1	0.57	42	220 (21.11 %)	0.34	118	16	12	0.09	1	6	0	0.31
			5	431	22	0.53	4	91 (8.73 %)	0.09	0	2	2	0.03	0	4	31	0.58
			10	341	0	0.05	1	44 (4.22 %)	0.05	0	0	0	0.01	0	0	0	0.05
Msteinb2	50	63	1	696	3	0.46	11	108 (10.36 %)	0.1	163	36	31	0.18	0	0	3	0.47
			5	600	1	0.22	15	137 (13.15 %)	0.28	12	16	22	0.1	0	2	1	0.22
			10	300	57	0.71	70	73 (7.01 %)	0.29	0	26	34	0.15	0	16	88	0.93
Msteinb3	50	63	1	1205	2	0.15	150	497 (47.70 %)	2.16	18	6	0	0.02	3	2	0	0.19
			5	924	38	0.44	57	230 (22.07 %)	0.7	56	20	32	0.13	2	2	35	0.44
			10	649	0	0.03	0	68 (6.53 %)	0.03	0	0	0	0.01	0	0	0	0.04
Msteinb4	50	100	1	455	11	7.98	97	291 (27.93 %)	0.96	16	8	0	0.03	436	48	6	10.94
			5	455	13	9.81	56	156 (14.97 %)	0.84	14	6	0	0.03	140	58	10	14.74
			10	455	3	1.81	33	113 (10.84 %)	0.78	3506	60	64	1.26	222	6	0	1.25
Msteinb5	50	100	1	666	2	2.2	93	277 (16.95 %)	0.88	0	0	0	0.01	1	14	5	5.73
			5	666	14	6.17	30	171 (10.47 %)	0.86	0	0	0	0.01	0	4	5	2.33
			10	652	0	0.15	5	88 (5.39 %)	0.16	0	0	0	0.02	0	0	0	0.15
Msteinb6	50	100	1	1269	15	10.35	80	319 (19.52 %)	0.57	1731	76	34	0.48	1218	14	1	7.06
			5	1262	136	10.17	666	711 (43.51 %)	14.04	171	22	6	0.19	2152	68	146	19.54
			10	903	28	4.14	113	194 (11.87 %)	1.38	4	4	5	0.06	4	16	38	3.65
Msteinb7	75	94	1	674	1	0.35	168	546 (34.93 %)	2.13	37	26	3	0.1	0	2	3	0.37
			5	662	0	0.16	2	135 (8.64 %)	0.27	2	2	0	0.03	0	0	0	0.16
			10	432	21	0.36	32	71 (4.54 %)	0.36	2	16	43	0.2	0	2	18	0.35
Msteinb8	75	94	1	836	0	0.19	120	394 (25.21 %)	1.3	9907	1586	12057	87.18	0	0	0	0.19
			5	761	44	0.76	441	856 (54.77 %)	15.98	2719	700	3842	22.79	6	24	36	0.86
			10	537	4	0.2	196	330 (21.11 %)	1.77	32	8	8	0.07	84	22	1	0.3
Msteinb9	75	94	1	1761	7	1.72	87	420 (26.87 %)	0.63	950	128	260	1.66	30	20	4	2.29
			5	1388	18	1.25	44	280 (17.91 %)	0.83	43	16	51	0.31	4	4	17	1.22
			10	816	115	2.35	110	204 (13.05 %)	0.86	4	24	82	0.32	0	26	122	2.66
Msteinb10	75	150	1	702	16	27.28	107	513 (20.86 %)	2.29	499046	9718	20907	[inf]	1086	36	38	74.31
			5	702	14	19.92	139	433 (17.61 %)	4.22	734	26	4	0.28	1997	58	15	83.27
			10	694	6	2.54	39	214 (8.70 %)	1.75	9	4	4	0.13	16	8	5	3.65
Msteinb11	75	150	1	893	37	36.03	188	590 (23.99 %)	4.05	34717	858	980	31.01	480	62	37	87.27
			5	893	3	9.36	112	787 (32.00 %)	7.84	5082	236	146	3.82	515	36	12	40.86
			10	855	112	11.72	364	619 (25.17 %)	8.62	138	14	1	0.1	874	10	0	1.25
Msteinb12	75	150	1	1867	26	21.96	94	421 (17.12 %)	1.17	10091	198	43	2.27	206	36	15	29.25
			5	1866	7	5.29	221	436 (17.73 %)	5.45	23904	554	399	12.35	1185	10	4	6.01
			10	1401	153	11.72	187	194 (7.89 %)	2.41	1247	48	181	2.18	111	30	197	17.06
Msteinb13	100	125	1	785	11	5.4	176	708 (33.97 %)	4.43	7708	946	1362	18.79	4	22	5	5.55
			5	745	28	1.71	156	265 (12.72 %)	2.19	12	10	12	0.19	3	10	44	1.88
			10	465	81	2.51	90	191 (9.17 %)	0.99	41	50	127	0.77	3	12	72	1.97
Msteinb14	100	125	1	1403	3	0.92	263	1219 (58.49 %)	26.97	61429	7476	15404	733.41	0	4	1	0.66
			5	1033	23	0.67	144	283 (13.58 %)	1.43	21	22	27	0.31	4	8	87	1.85
			10	595	56	1.64	16	66 (3.17 %)	0.17	0	10	13	0.11	0	4	10	0.36
Msteinb15	100	125	1	2555	10	3.13	63	529 (25.38 %)	1.07	5924	476	136	5.59	1	4	17	4.68
			5	1891	0	0.17	1	154 (7.39 %)	0.17	4	0	0	0.08	0	0	0	0.17
			10	1086	117	2.74	30	113 (5.42 %)	0.49	6	12	27	0.25	0	24	83	2.89
Msteinb16	100	200	1	840	104	141.69	721	1349 (41.08 %)	38.78	12	2	4	0.11	1462	58	117	291.61
			5	840	21	57.35	334	410 (12.48 %)	7.08	12	2	4	0.13	2804	116	60	412.36
			10	800	44	26.92	720	772 (23.51 %)	21.41	17424	446	438	67.77	6298	118	20	49.92
Msteinb17	100	200	1	1299	25	33.48	655	1621 (49.36 %)	69.3	643509	10274	20590	[inf]	421	26	6	21.32
			5	1299	28	33.54	636	1264 (38.49 %)	106.78	354342	11374	7695	[inf]	2548	38	16	88.87
			10	1178	67	8.57	2040	1472 (44.82 %)	115.28	151537	6244	14442	[3.5%]	412	142	113	18.94
Msteinb18	100	200	1	2585	16	53.49	393	1123 (34.20 %)	18.28	168041	2408	4411	212.09	1202	46	39	146.2
			5	2585	36	46.8	337	839 (25.55 %)	25.35	322152	5480	2352	1439.86	24	16	6	21.6
			10	1997	22	7.11	237	547 (16.66 %)	10.88	5108	122	142	3.48	220	10	34	8.39

Table 7: Results - Instances MStein, $h = 9$

Instance					A2		A3			A4				A7			
Name	n	|E|	h	Opt	B&B nodes	Seconds	B&B nodes	Cuts (%)	Seconds	B&B nodes	Sep	Hops	Seconds	B&B nodes	Sep	Hops	Seconds
Mstеinb1	50	63	1	467	6	1.67	13	161 (11.39 %)	0.22	0	0	0	0	0	10	1	1.26
			5	431	25	0.9	4	100 (7.07 %)	0.13	0	2	2	0.03	0	2	21	0.94
			10	341	0	0.07	0	44 (3.11 %)	0.04	0	0	0	0.01	0	0	0	0.07
Mstеinb2	50	63	1	696	6	2.2	21	228 (16.12 %)	0.3	9	6	1	0.03	8	14	1	1.85
			5	600	17	0.61	32	201 (14.21 %)	0.37	0	22	17	0.1	0	6	16	0.95
			10	300	74	1.22	75	80 (5.66 %)	0.29	0	26	34	0.15	0	32	39	1.18
Mstеinb3	50	63	1	1205	6	1.38	46	316 (22.35 %)	0.35	6	4	0	0.01	0	4	0	0.34
			5	931	24	1.06	54	312 (22.07 %)	0.77	24	8	24	0.08	0	10	21	0.81
			10	649	0	0.03	0	75 (5.30 %)	0.04	0	0	0	0.01	0	0	0	0.04
Mstеinb4	50	100	1	455	17	19.42	46	286 (20.23 %)	0.61	7	4	0	0.02	519	62	11	39.59
			5	455	21	18.88	36	163 (11.53 %)	0.8	7	4	0	0.03	1404	28	6	25.31
			10	455	1	1.53	40	166 (11.74 %)	1.05	933	44	42	0.75	1018	12	2	5.89
Mstеinb5	50	100	1	666	10	14.48	72	199 (8.93 %)	0.53	0	0	0	0.01	1390	46	11	38.95
			5	666	11	14.48	76	314 (14.09 %)	1.17	0	0	0	0.02	248	44	12	23.7
			10	652	1	0.8	3	121 (5.43 %)	0.2	0	0	0	0.02	0	0	1	0.81
Mstеinb6	50	100	1	1269	19	27.66	114	395 (17.73 %)	0.91	42	16	4	0.08	1090	20	1	17.59
			5	1262	928	105.14	735	992 (44.52 %)	26.37	1	4	0	0.03	2253	88	306	82.21
			10	903	144	22.04	9	199 (8.93 %)	0.41	0	6	10	0.09	10	34	114	27.18
Mstеinb7	75	94	1	674	14	4.49	283	1054 (49.69 %)	16.23	6	6	2	0.04	52	60	10	5.49
			5	662	14	1.8	4	128 (6.03 %)	0.29	0	2	0	0.02	0	22	13	1.52
			10	432	50	1.96	50	59 (2.78 %)	0.35	0	16	41	0.19	0	40	60	2.07
Mstеinb8	75	94	1	836	2	0.8	209	801 (37.77 %)	5.69	762	106	29	0.54	3	30	3	1.86
			5	832	3	0.52	238	752 (35.45 %)	6.52	12	4	0	0.03	5	16	5	0.81
			10	537	70	2.16	227	427 (20.13 %)	2.49	8	44	27	0.18	30	22	22	0.85
Mstеinb9	75	94	1	1761	15	6.3	91	535 (25.22 %)	1.03	163	14	8	0.16	2	24	8	6.84
			5	1388	83	4.16	152	444 (20.93 %)	2.03	15	4	9	0.1	3	4	20	2.24
			10	816	104	7.33	154	279 (13.15 %)	1.02	3	24	88	0.3	0	22	104	3.94
Mstеinb10	75	150	1	702	11	42.5	121	509 (15.18 %)	3	8593	88	7	1.17	3518	54	19	234.81
			5	702	10	38.73	108	460 (13.72 %)	2.48	3974	92	51	2.75	1394	32	0	52.88
			10	694	105	71.31	13	212 (6.32 %)	0.91	1	4	3	0.09	703	28	23	27.24
Mstеinb11	75	150	1	893	15	68.13	236	645 (19.24 %)	3.11	9612	208	55	2.83	1447	22	2	41.53
			5	893	22	79.01	49	270 (8.05 %)	1.41	3801	46	8	1.38	548	38	16	73.69
			10	855	349	113.03	1144	991 (29.56 %)	45.2	183	12	2	0.11	1190	26	8	13.95
Mstеinb12	75	150	1	1867	21	75	159	651 (19.42 %)	3.66	1915	58	19	0.64	2176	56	4	133.38
			5	1866	4	18.92	14	389 (11.60 %)	1.7	0	0	0	0.04	571	14	4	27.86
			10	1401	108	14.59	31	232 (6.92 %)	0.53	10	8	79	0.47	27	2	97	15.5
Mstеinb13	100	125	1	785	19	13.17	64	488 (17.26 %)	0.98	179	46	3	0.31	4	18	2	6.68
			5	745	31	4.87	19	255 (9.02 %)	1.02	0	10	12	0.18	1	26	20	5.54
			10	465	90	6.05	102	268 (9.48 %)	1.22	6	42	87	0.58	0	8	64	3.85
Mstеinb14	100	125	1	1403	24	7.33	187	827 (29.24 %)	4.69	24	10	6	0.17	69	40	10	10.67
			5	1038	102	5.58	112	303 (10.71 %)	1.49	5	44	87	0.69	0	32	92	5.55
			10	595	75	5.24	18	52 (1.84 %)	0.16	0	10	13	0.11	0	46	68	6.45
Mstеinb15	100	125	1	2555	19	14.59	64	739 (26.13 %)	1.59	134700	8388	22809	1736.98	10	40	11	17.74
			5	1891	0	0.26	2	186 (6.58 %)	0.21	0	0	0	0.03	0	0	0	0.28
			10	1109	4	0.85	6	111 (3.93 %)	0.3	0	8	19	0.2	0	6	5	0.81
Mstеinb16	100	200	1	840	11	125.86	838	2403 (53.66 %)	347.61	3	0	1	0.07	35	10	3	33.36
			5	840	24	163.98	395	751 (16.77 %)	13.91	3	0	1	0.07	35	10	3	33.35
			10	800	15	65.88	523	1280 (28.58 %)	62.78	2098	28	11	0.73	2011	32	10	62.23
Mstеinb17	100	200	1	1299	40	164.15	1153	3039 (67.87 %)	1201.42	15540	250	103	6.26	3620	66	3	174.34
			5	1299	22	95.57	807	2098 (46.85 %)	249.64	28696	760	471	32.26	1443	32	8	101.9
			10	1225	13	15.85	13	242 (5.40 %)	1.47	2650	118	76	3.67	4	4	14	15.23
Mstеinb18	100	200	1	2585	36	245.45	164	553 (12.35 %)	2.07	17636	270	36	5.35	2692	90	5	490.44
			5	2585	36	140.63	326	966 (21.57 %)	16.83	3632	46	20	2.34	3107	40	2	171.13
			10	1997	511	168.49	592	786 (17.55 %)	21.28	15634	432	421	22.77	2701	94	479	273.81

Table 8: Results - Instances MStein, $h = 12$

Instance					A2		A3			A4				A7			
Name	n	$\|E\|$	δ	Opt	B&B nodes	Seconds	B&B nodes	Cuts (%)	Seconds	B&B nodes	Sep	Hops	Seconds	B&B nodes	Sep	Hops	Seconds
steinc1_10	500	625	1	8	0	0.13	501	2169 (39.49 %)	455.95	14935	4158	3175	[inf]	0	0	0	0.14
steinc1_100			1	71	0	0.13	426	2207 (40.19 %)	347.88	12006	3668	3724	[inf]	0	0	0	0.14
steinc1_10			10	8	0	0.12	265	1266 (23.05 %)	88.19	15154	3746	3035	[inf]	0	0	0	0.14
steinc1_100			10	71	0	0.12	246	1310 (23.85 %)	77.98	14909	3380	2501	[inf]	0	0	0	0.13
steinc1_10			30	8	0	0.12	316	1352 (24.62 %)	161.06	7499	2902	5551	[inf]	0	0	0	0.14
steinc1_100			30	71	0	0.12	334	1302 (23.71 %)	205.27	6956	2789	5667	[74%]	0	0	0	0.13
steinc2_10			1	32	0	0.12	24	1436 (26.15 %)	60.22	33072	5442	2770	[inf]	0	0	0	0.16
steinc2_100			1	328	0	0.12	23	1427 (25.98 %)	58.42	35917	5992	3302	[inf]	0	0	0	0.16
steinc2_10			10	32	0	0.12	230	2032 (37.00 %)	262.91	14880	4384	3629	[inf]	0	0	0	0.15
steinc2_100			10	328	0	0.13	231	2064 (37.58 %)	260.33	29292	5242	3960	[inf]	0	0	0	0.16
steinc2_10			30	32	0	0.13	140	1173 (21.36 %)	76.2	11110	5251	14760	[inf]	0	0	0	0.16
steinc2_100			30	328	0	0.13	255	1233 (22.45 %)	158.22	12916	6512	14632	[inf]	0	0	0	0.15
steinc3_10			1	151	0	0.13	108	2523 (45.94 %)	72.54	188193	14358	3496	[inf]	0	0	0	0.24
steinc3_100			1	1519	0	0.13	152	2719 (49.51 %)	60.75	200501	14542	2211	[inf]	0	0	0	0.24
steinc3_10			10	151	0	0.13	2	2098 (38.20 %)	25.61	100891	17263	3116	[inf]	0	0	0	0.25
steinc3_100			10	1519	0	0.13	3	2138 (38.93 %)	23	109759	16118	3053	[inf]	0	0	0	0.24
steinc3_10			30	95	23	0.24	921	1387 (25.25 %)	119.8	67075	9275	5601	[11%]	0	0	23	1.87
steinc3_100			30	968	32	0.26	1218	1682 (30.63 %)	170.03	83343	11429	4524	[17%]	0	0	32	1.85
steinc4_10			1	115	0	0.13	0	2521 (45.90 %)	18.18	107664	13082	2554	[inf]	0	0	0	0.2
steinc4_100			1	1148	0	0.12	0	2568 (46.76 %)	17.9	130870	13042	2210	[inf]	0	0	0	0.2
steinc4_10			10	115	0	0.13	0	2279 (41.50 %)	25.35	80371	16477	1893	[92%]	0	0	0	0.2
steinc4_100			10	1148	0	0.12	0	2355 (42.88 %)	27.65	86901	17311	1951	[inf]	0	0	0	0.2
steinc4_10			30	84	12	0.18	1023	1671 (30.43 %)	148.86	50117	14057	3417	[78%]	0	0	12	0.89
steinc4_100			30	854	31	0.2	587	1552 (28.26 %)	128.09	48141	12612	3539	[68%]	0	0	31	1.98
steinc5_10			1	258	0	0.13	0	3165 (57.63 %)	11.4	268429	13146	591	[inf]	0	0	0	0.26
steinc5_100			1	2600	0	0.12	0	3190 (58.08 %)	12.75	187660	12592	809	[inf]	0	0	0	0.26
steinc5_10			10	258	0	0.13	2	2629 (47.87 %)	32.59	179599	16173	776	[inf]	0	0	0	0.26
steinc5_100			10	2600	0	0.12	0	2573 (46.85 %)	32.84	201338	15693	407	[inf]	0	0	0	0.26
steinc5_10			30	154	0	0.13	392	864 (15.73 %)	53.04	105420	15395	2501	[23%]	0	0	0	0.26
steinc5_100			30	1584	0	0.12	417	1284 (23.38 %)	74.19	79122	14671	3863	[15%]	0	0	0	0.19

Table 9: Results - Instances steinc, $h = 5$

Instance					A2		A3			A4				A7			
Name	n	$\|E\|$	h	Opt	B&B nodes	Seconds	B&B nodes	Cuts (%)	Seconds	B&B nodes	Sep	Hops	Seconds	B&B nodes	Sep	Hops	Seconds
steinc1_10	500	625	1	27	184	[inf]	2139	3278 (18.24 %)	**1596.01**	60077	4080	1783	[inf]	75	108	39	3969.16
steinc1_100			1	274	165	[inf]	2295	4146 (23.07 %)	**1883.9**	74007	3880	1643	[inf]	132	158	67	6656.76
steinc1_10			10	27	179	6286.32	237	494 (2.75 %)	**65.24**	21384	3366	3108	[inf]	57	86	39	2505.47
steinc1_100			10	274	31	1493.17	237	494 (2.75 %)	**64.51**	22378	3202	3640	[inf]	41	124	80	3833.13
steinc1_10			30	27	5	191.45	97	272 (1.51 %)	12.85	5	2	4	**1.16**	0	20	0	153.31
steinc1_100			30	274	5	191.24	57	306 (1.70 %)	9.8	5	2	4	**1.18**	0	20	0	153.26
steinc2_10			1	59	138	[inf]	1114	6037 (33.59 %)	[inf]	11421	1008	346	**350.6**	605	244	72	[inf]
steinc2_100			1	604	205	[inf]	1219	5969 (33.21 %)	[inf]	11421	1008	346	**350.87**	423	344	116	[inf]
steinc2_10			10	59	30	**918.91**	819	5684 (31.63 %)	[inf]	31601	5060	4031	[inf]	150	190	16	3377.29
steinc2_100			10	604	30	**914.8**	1402	4406 (24.52 %)	4389.98	35774	5376	3909	[inf]	187	214	50	3802.01
steinc2_10			30	53	948	4883.41	29	276 (1.54 %)	6.73	1	40	18	**5.98**	21	72	81	723.36
steinc2_100			30	546	726	3476.54	110	351 (1.95 %)	25.16	1110	38	8	**6.21**	127	349	1633	7055.94
steinc3_10			1	439	70	**2452.75**	777	6812 (37.90 %)	[inf]	580833	13260	1569	7105.42	49	228	55	[inf]
steinc3_100			1	4463	71	**2702.21**	803	7167 (39.88 %)	[inf]	644089	14220	988	[inf]	185	208	35	5940.32
steinc3_10			10	289	343	2991.51	1556	2362 (13.14 %)	217.64	948	202	364	**96.52**	98	24	18	290.79
steinc3_100			10	2971	953	5862.92	1588	2123 (11.81 %)	**314.37**	10452	1831	1368	571.35	120	46	254	1388.61
steinc3_10			30	129	2	2.38	303	399 (2.22 %)	24.71	10	6	3	**0.78**	4	6	6	6.32
steinc3_100			30	1343	0	1.51	0	251 (1.40 %)	0.85	0	4	0	**0.38**	0	4	0	1.88
steinc4_10			1	648	81	**2309.83**	818	7402 (41.19 %)	[inf]	712947	12868	521	[inf]	37	168	56	4660.12
steinc4_100			1	6566	89	**1945.2**	1050	7792 (43.36 %)	[inf]	755420	12166	528	[inf]	124	100	46	3409.11
steinc4_10			10	336	86	520.6	2017	3070 (17.08 %)	1567.34	123295	14585	2501	[inf]	10	16	30	**103.09**
steinc4_100			10	3458	158	**277.06**	3405	3742 (20.82 %)	2408.75	38232	7884	2651	4509.2	103	56	226	763.35
steinc4_10			30	134	133	306.11	655	824 (4.58 %)	43.65	216	182	228	**43.55**	0	46	28	115.6
steinc4_100			30	1380	343	1190.37	1377	1416 (7.88 %)	**133.72**	1224	852	442	179.41	6	144	414	1797.69
steinc5_10			1	1248	108	**2938.07**	719	8718 (48.51 %)	[inf]	864230	7754	2886	[inf]	21	84	73	3413.95
steinc5_100			1	12533	77	**2127.61**	542	8270 (46.02 %)	[inf]	772432	8114	2542	[inf]	46	66	78	3690.84
steinc5_10			10	494	138	**234.05**	3171	3584 (19.94 %)	956.06	51045	4650	882	1361.39	829	38	53	334.16
steinc5_100			10	5032	270	480.62	2860	2817 (15.67 %)	1247.7	57725	5297	1064	1790.83	665	14	17	**180.13**
steinc5_10			30	182	66	233.66	455	1062 (5.91 %)	52.91	210	58	7	**5.91**	153	64	22	167.4
steinc5_100			30	1857	152	388.55	511	1365 (7.60 %)	**45.49**	1083	380	146	53.26	234	52	95	297.16

Table 10: Results - Instances steinc, $h = 15$

Instance					A2		A3			A4				A7			
Name	n	$\|E\|$	h	Opt	B&B nodes	Seconds	B&B nodes	Cuts (%)	Seconds	B&B nodes	Sep	Hops	Seconds	B&B nodes	Sep	Hops	Seconds
steinc1_10	500	625	1	27	22	[inf]	2508	5404 (17.75 %)	3916.94	53385	1822	697	[inf]	43	72	8	[inf]
steinc1_100			1	-	26	[inf]	3563	7554 (24.81 %)	[inf]	53522	1828	700	[inf]	42	72	8	[inf]
steinc1_10			10	-	30	[inf]	1359	7215 (23.09 %)	[inf]	21024	2456	2867	[inf]	0	92	0	[inf]
steinc1_100			10	-	25	[inf]	1271	7157 (23.50 %)	[inf]	21112	2176	2357	[inf]	0	96	0	[inf]
steinc1_10			30	27	12	2190.41	73	172 (0.56 %)	10.46	0	2	0	0.31	0	54	6	3321.67
steinc1_100			30	274	12	2206.83	73	172 (0.56 %)	9.51	0	2	0	0.31	0	42	7	2604.59
steinc2_10			1	59	30	[inf]	3228	6722 (22.07 %)	5523.22	0	30	0	1.96	1001	120	0	[inf]
steinc2_100			1	604	33	[inf]	1098	3239 (10.64 %)	368.72	0	30	0	1.96	1007	104	1	[inf]
steinc2_10			10	59	21	3811.23	100	976 (3.21 %)	26.88	0	30	0	1.95	599	122	0	[inf]
steinc2_100			10	604	57	[4%]	100	976 (3.21 %)	26.51	0	30	0	1.95	599	122	0	[inf]
steinc2_10			30	53	46	[inf]	173	749 (2.46 %)	27.19	0	40	18	5.92	13	78	44	[12%]
steinc2_100			30	546	46	[inf]	198	387 (1.27 %)	24.13	1001	34	5	5.38	0	48	37	[inf]
steinc3_10			1	439	18	[inf]	554	8558 (28.10 %)	[inf]	24595	632	56	235.02	0	62	0	[inf]
steinc3_100			1	4463	19	[inf]	492	8361 (27.46 %)	[inf]	24595	632	56	235.09	0	74	0	[inf]
steinc3_10			10	289	35	[inf]	348	2197 (7.21 %)	68.44	6	74	108	30.37	6	32	58	3297.52
steinc3_100			10	2979	51	[inf]	723	2978 (9.78 %)	253.88	3	40	170	26.07	0	54	92	[2%]
steinc3_10			30	129	2	13.21	5	265 (0.87 %)	1.45	0	6	3	0.8	0	6	1	6.95
steinc3_100			30	1343	18	20.42	0	250 (0.82 %)	0.94	0	4	0	0.3	0	4	0	3.11
steinc4_10			1	-	15	[inf]	631	9605 (31.54 %)	[inf]	660344	4968	1103	[inf]	0	44	7	[inf]
steinc4_100			1	-	21	[inf]	862	9880 (32.44 %)	[inf]	814417	6766	1157	[inf]	0	38	22	[inf]
steinc4_10			10	341	85	4503.77	154	1672 (5.49 %)	25.72	0	4	3	2.25	2	8	4	990.38
steinc4_100			10	3504	98	[inf]	685	2757 (9.05 %)	226.6	0	4	0	0.85	0	4	1	102.46
steinc4_10			30	136	40	[10%]	19	661 (2.17 %)	5.34	0	10	5	2.35	0	10	44	2665.68
steinc4_100			30	1396	61	[inf]	321	866 (2.84 %)	62.29	0	6	5	2.1	0	14	36	980.53
steinc5_10			1	1248	22	[inf]	498	11505 (37.78 %)	[inf]	0	0	0	0.86	0	22	6	[inf]
steinc5_100			1	12533	19	[inf]	589	11120 (36.52 %)	[inf]	0	0	0	0.86	0	36	5	[inf]
steinc5_10			10	495	74	[inf]	3558	4938 (16.22 %)	3401.39	1218	46	58	19.7	923	30	25	5416.15
steinc5_100			10	5044	50	[inf]	1299	2800 (9.19 %)	1478.63	435	24	25	10.16	130	42	87	6906.24
steinc5_10			30	183	43	[inf]	42	1022 (3.36 %)	16.89	10	128	136	28.45	10	34	74	[3%]
steinc5_100			30	1860	34	[inf]	86	1060 (3.48 %)	15.67	0	18	86	7.69	0	12	63	266.11

Table 11: Results - Instances steinc, $h = 25$

References

Beasley, J. E. (1990). OR-Library: distributing test problems by electronic mail, *Journal of the Operational Research Society* **41**: 1069–1072.

Bienstock, D., Goemans, M. X., Simchi-Levi, D. and Williamson, D. (1993). A note on the prize collecting traveling salesman problem, *Mathematical Programming* **59**: 413–420.

Canuto, S. A., Resende, M. G. C. and Ribeiro, C. C. (2001). Local search with perturbations for the prize-collecting Steiner tree problem in graphs, *Networks* **38**: 50–58.

Chopra, S. and Rao, M. R. (1994). The Steiner tree problem I: Formulations, compositions and extension of facets, *Mathematical Programming* **64**: 209–229.

Cole, R., Hariharan, R., Lewenstein, M. and Porat, E. (2001). A faster implementation of the Goemans-Williamson clustering algorithm, SODA'01: Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms, Washington : 17–25.

Costa, A. M., Cordeau, J.-F. and Laporte, G. (2006). Steiner tree problems with profits, *INFOR* **44**: 99–115.

Courant, R. and Robbins, H. (1941). *What is mathematics?*, Oxford University Press. New York.

Cunha, A. S., Lucena, A., Maculan, N. and Resende, M. G. C. (2003). A relax and cut algorithm for the prize collecting Steiner problem in graphs, Proceedings of Mathematical Programming in Rio, Búzios - Brazil : 72–78.

Dahl, G., Gouveia, L. and Requejo, C. (2006). On formulations and methods for the hop-constrained minimum spanning tree problem, *in* P. Pardalos and M. G. C. Resende (eds), *Handbooks of Telecommunications*.

Dantzig, G. B., Fulkerson, D. R. and Johnson, S. M. (1954). Solution of a large-scale traveling salesman problem, *Operations Research* **2**: 393–410.

Fischetti, M. (1991). Facets of two Steiner arborescence polyhedra, *Mathematical Programming* **51**: 401–419.

Garey, M. R., Graham, R. L. and Johnson, D. S. (1977). The complexity of computing Steiner minimal trees, *SIAM Journal on Applied Mathematics* **32**: 835–859.

Gilbert, E. N. and Pollak, H. O. (1968). Steiner minimal trees, *SIAM Journal on Applied Mathematics* **16**: 1–29.

Goemans, M. X. (1994). The Steiner tree polytope and related polyhedra, *Mathematical Programming* **63**: 157–182.

Goemans, M. X. and Williamson, D. P. (1995). A general approximation technique for constrained forest problems, *SIAM Journal on Computing* **24**: 296–317.

Gouveia, L. (1995). Using the Miller-Tucker-Zemlin constraints to formulate a minimal spanning tree problem with hop constraints, *Computers & Operations Research* **22**: 959–970.

Gouveia, L. (1996). Multicommodity flow models for spanning trees with hop constraints, *European Journal of Operational Research* **95**: 178–190.

Gouveia, L. (1998). Using variable redefinition for computing lower bounds for minimum spanning and Steiner trees with hop constraints, *INFORMS Journal on Computing* **10**: 180–188.

Gouveia, L. and Requejo, C. (2001). A new Lagrangean relaxation approach for the hop-constrained minimum spanning tree problem, *European Journal of Operational Research* **132**: 539–552.

Hwang, F. K., Richards, D. S. and Winter, P. (1992). *The Steiner Tree Problem*, North-Holland, Amsterdan.

Johnson, D. S., Minkoff, M. and Phillips, S. (2000). The prize collecting Steiner tree problem: theory and practice, Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms, San Francisco : 760–769.

Khoury, B. N., Pardalos, P. M. and Du, D.-Z. (1993). A test problem generator for the Steiner tree problem in graphs, *ACM Transactions on Mathematical Software* **19**: 509–522.

Klau, G. W., Ljubić, I., Moser, A., Mutzel, P., Neuner, P., Pferschy, U., Raidl, G. and Weiskircher, R. (2004). Combining a memetic algorithm with integer programming to solve the prize-collecting Steiner tree problem, *Springer Lecture Notes in Computer Science* **3102**: 1304–1315.

Koch, T. and Martin, A. (1998). Solving Steiner tree problems in graphs to optimality, *Networks* **32**: 207–232.

Ljubić, I., Weiskircher, R., Pferschy, U., Klau, G., Mutzel, P. and Fischetti, M. (2005). Solving the prize-collecting Steiner tree problem to optimality, Proceedings of ALENEX, Seventh Workshop on Algorithm Engineering and Experiments, Vancouver, Canada .

Lucena, A. (1991). Tight bounds for the Steiner problem in graphs, Presentation made at the TIMS XXX - SOBRAPO XXIII Joint International Meeting, Rio de Janeiro, Brazil.

Lucena, A. and Resende, M. G. C. (2004). Strong lower bounds for the prize collecting Steiner problem in graphs, *Discrete Applied Mathematics* **141**: 277–294.

Margot, F., Prodon, A. and Liebling, T. M. (1994). Tree polyhedron on 2-tree, *Mathematical Programming* **63**: 183–192.

Miller, C. E., Tucker, A. W. and Zemlin, R. A. (1960). Integer programming formulations and travelling salesman problems, *Journal of the Association of Computing Machinery* **7**: 326–329.

Segev, A. (1987). The node-weighted Steiner tree problem, *Networks* **17**: 1–17.

Voß, S. (1999). The Steiner tree problem with hop constraints, *Annals of Operations Research* **86**: 321–345.